QingKeV2 マイクロプロセッサマニュアル

V1.3

概要

QingKe V2シリーズマイクロプロセッサは、標準RISC-V命令セットRV32IのサブセットRV32Eに基づく32ビット汎用MCUマイクロプロセッサであり、汎用レジスタは16個のみで、RV32Iの半分となっている。これにより、より簡素化された組込み用途に適している。V2シリーズは標準RV32EC命令拡張をサポートし、V2Cはハードウェア乗算をサポートする。さらに、カスタムXW拡張、Hardware Prologue/Epilogue(HPE)、Vector Table Free(VTF)、より簡素化された1線/2線シリアルデバッグインタフェース(SDI)、および「WFE」命令を備えている。

機能

Features	Description
Instruction Set Architecture (ISA)	RV32EmC
Pipeline	2
Branch prediction	静的分岐予測
Interrupt	例外を含む合計256個の割り込みをサポートし、VTFをサポートする
Hardware Prologue/Epilogue (HPE)	2段のハードウェア割込スタック
Low-power consumption mode	スリープモードおよびディープスリープモードをサポートし、WFIおよびWFEスリープ方式をサポートする
Extended Instruction Set	ハーフワードおよびバイト操作の縮小命令をサポートする
Debug	1線／2線SDI、標準RISC-Vデバッグ

Chapter 1 概要

QingKe V2シリーズマイクロプロセッサは、RISC-Vアーキテクチャに基づき、RV32ECサブセットに準拠した32ビットMCUマイクロプロセッサである。汎用レジスタは16個のみで、より簡素化された構造を持ち、組込み用途における面積および消費電力の要件に適している。主な機能は以下の表1-1に示す。

表1-1 QingKe V2マイクロプロセッサの概要

Feature / Model	ISA	HPE number of levels	Interruptions nesting number of levels	VTF number of channels	Pipeline	Vector table mode	Extended Instruction (XW)	Number of memory protection areas
V2A	RV32EC	2	2	2	2	Address/ Instruction	√	×
V2C	RV32EmC	2	2	2	2	Address/ Instruction	√	×

注:OSのタスク切り替えでは通常スタックへのプッシュが使用され、レベル数に制限はない。

1.1 命令セット

QingKe V2シリーズマイクロプロセッサは、標準RV32EC命令セットアーキテクチャ(ISA)に準拠している。標準の詳細な文書は、RISC-V Internationalのウェブサイトにある「The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Document Version 2.2」に記載されている。RISC-V命令セットは単純なアーキテクチャを持ち、モジュール設計をサポートしており、異なるニーズに応じた柔軟な組み合わせが可能である。QingKe V2シリーズマイクロプロセッサがサポートする命令セット拡張は以下のとおりである。

- RV32:32ビットアーキテクチャ、汎用レジスタのビット幅は32ビット
- E:RV32Iサブセット、汎用レジスタは16個のみサポート
- C:16ビット縮小命令をサポート
- XW:バイトおよびハーフワード操作の自己拡張用16ビット縮小命令
- m:ハードウェア乗算、すなわちZmmul拡張

注:

1. コード密度をさらに向上させるため、XWサブセットを以下の圧縮命令c.lbu/c.lhu/c.sb/c.sh/c.lbusp/c.lhusp/c.sbsp/c.shspで拡張する。使用にはMRSコンパイラまたはその提供するツールチェーンが必要である。
2. V2Cにおけるm拡張はハードウェア乗算命令、すなわちZmmul拡張のみを含み、使用にはMRSコンパイラまたはその提供するツールチェーンが必要である。

1.2 レジスタセット

RV32EはRV32Iのサブセットであり、レジスタ数は半分の x0～x15までの16個のレジスタセットが存在する。以下の表1-2にRV32Eのレジスタとその説明を示す。

表1-2 RV32Eレジスタ

Register	ABI Name	Description	Storer
x0	zero	Hardcoded 0	-
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	-
x4	tp	Thread pointer	-
x5-7	t0-2	Temporary register	Caller
x8	s0/fp	Save register/frame pointer	Callee
x9	s1	Save register	Callee
x10-11	a0-1	Function parameters/return values	Caller
x12-15	a2-5	Function parameters	Caller

上記の表におけるCaller属性は、呼び出された手続きがレジスタ値を保存しないことを意味し、Callee属性は、呼び出された手続きがレジスタを保存することを意味する。

1.3 特権モード

標準RISC-Vアーキテクチャには、Machineモード、Supervisorモード、Userモードの3つの特権モードが含まれており、以下の表1-3に示す。

Machineモードは必須モードであり、他のモードはオプションである。詳細については、RISC-V Internationalのウェブサイトから無償でダウンロード可能な「The RISC-V Instruction Set Manual Volume II: Privileged Architecture」を参照のこと。

表1-3 RISC-Vアーキテクチャの特権モード

Code	Name	略語
0b00	User Mode	U
0b01	Supervisor Model	S
0b10	Reserved	Reserved
0b11	Machine mode	M

Machineモードは最も高い特権を持ち、このモードではプログラムがすべての制御およびステータスレジスタ（CSR）にアクセス可能である。また、QingKe V2にはPhysical Memory Protection（PMP）ユニットは設計されていない。

CSRレジスタmstatus（Machineモードステータスレジスタ）内のMPPビットはデフォルトで0b11であり、常にMachineモードで動作していることを示す。

1.4 CSRレジスタ

RISC-Vアーキテクチャでは、マイクロプロセッサの動作状態を制御および記録するために一連のCSRレジスタが定義されている。これらのCSRは、内部専用の12ビットアドレス符号空間を使用して最大4096個まで拡張可能である。CSR[11:10]の上位2ビットを使用して当該レジスタの読出し／書込み許可を定義し、0b00、0b01、0b10は読出し／書込み許可、0b11は読出し専用を示す。CSR[9:8]の2ビットを使用して、このレジスタにアクセス可能な最下位の特権レベルを定義し、その値は表1-3で定義された特権モードに対応する。標準で定義された関連CSRレジスタに加えて、QingKe V2シリーズマイクロプロセッサでは、拡張機能の制御およびステータス記録のためにいくつかのカスタムCSRレジスタが拡張されている。マイクロプロセッサにより実装されているCSRレジスタの詳細は第7章に記載されている。

Chapter 2 例外

例外機構とは、「異常な動作イベント」を捕捉して処理するための機構である。QingKe V2シリーズマイクロプロセッサは、割り込みを含む最大256個の例外を処理可能な例外応答システムを備えている。例外または割り込みが発生した際、マイクロプロセッサは迅速に応答し、当該例外および割り込みイベントを処理することができる。

2.1 例外の種類

例外または割り込みが発生した場合、マイクロプロセッサは現在のプログラムを中断し、例外または割り込みハンドラに移行し、処理完了後に中断されたプログラムへ復帰する。広義には、割り込みも例外の一部と見なされる。

現在発生している事象が割り込みか例外かは、Machineモードの例外原因レジスタmcauseを通じて判別可能である。mcause[31]は割り込みフィールドであり、例外の原因が割り込みか例外かを示すために使用される。mcause[31]==1は割り込み、mcause[31]==0は例外を意味する。mcause[30:0]は例外コードであり、例外の具体的な原因または割り込み番号を示すために使用される。詳細は以下の表2-1に示す。

表2-1 V2マイクロプロセッサの例外コード

Interrupt	Exception codes	Synchronous / Asynchronous	Reason for exception
1	0-1	-	Reserved
1	2	Precise asynchronous	NMI interrupts
1	3-11	-	Reserved
1	12	Precise asynchronous	SysTick interrupts
1	13	-	Reserved
1	14	Synchronous	Software interrupts
1	15	-	Reserved
1	16-255	Precise asynchronous	External interrupt 16-255
0	0	Synchronous	Instruction address misalignment
0	1	Synchronous	Fetch command access error
0	2	Synchronous	Illegal instructions
0	3	Synchronous	Breakpoints
0	4	Synchronous	Load instruction access address misalignment
0	5	Non- precision asynchronous	Load command access error
0	6	Synchronous	Store/AMO instruction access address misalignment
0	7	Non-precision asynchronous	Store/AMO command access error
0	8	Synchronous	Environment call in User mode (not supported in V2)
0	11	Synchronous	Environment call in Machine mode

表中の「同期(Synchronous)」とは、例外が発生した命令を正確に特定できることを意味し、例えばebreak命令やecall命令など、命令の実行ごとに例外が発生する場合が該当する。

「非同期(Asynchronous)」とは、例外発生時に命令を特定することができず、例外発生のたびに命令のPC値が異なる可能性があることを意味する。

「厳密な非同期(Precise asynchronous)」とは、例外が命令の境界、すなわち命令の実行完了後の状態で正確に特定できることを意味し、外部割り込みなどが該当する。

「非厳密な非同期(Non-precision asynchronous)」とは、命令の境界を正確に特定できず、命令の実行途中で中断された状態などが該当する。例えばメモリアクセスエラーなどである。メモリアクセスには時間がかかるため、マイクロプロセッサは通常、アクセス完了を待たずに命令の実行を継続する。アクセスエラー例外が再度発生した時点では、すでに後続の命令が実行されており、正確な位置を特定することができない。

2.2 例外への遷移

　プログラムが通常動作中に、何らかの理由で例外または割り込みが発生した場合、マイクロプロセッサのハードウェア動作は以下のように要約される。

(1) 現在のプログラムフローを中断し、例外または割り込み処理関数の実行に移行する。

　例外または割り込み処理関数のエントリベースアドレスおよびアドレッシング方式は、例外エントリベースアドレスレジスタmtvecによって定義される。mtvec[31:2]は例外または割り込み処理関数のベースアドレスを定義する。mtvec[1:0]はハンドラ関数のアドレッシング方式を定義し、そのうちmtvec[0]は例外および割り込みのエントリ方式を定義する。

　mtvec[0]==0の場合、すべての例外および割り込みは統一エントリを使用し、例外または割り込みが発生すると、mtvec[31:2]で定義されたベースアドレスに遷移して実行される。

　mtvec[0]==1の場合、例外および割り込みはベクタテーブル方式を使用し、各例外および割り込みには番号が割り当てられ、アドレスは割り込み番号×4だけシフトされる。例外または割り込みが発生すると、mtvec[31:2] + 割り込み番号×4のアドレスに遷移して実行される。

　ベクタモードにおけるmtvec[1]は、ベクタテーブルの識別方式を定義する。mtvec[1]==0の場合、ベクタテーブルに格納される命令は、例外または割り込み処理関数へのジャンプ命令、または他の任意の命令である。mtvec[1]==1の場合、ベクタテーブルには例外処理関数の絶対アドレスが格納される。

　なお、QingKe V2マイクロプロセッサでは、ベクタテーブルのベースアドレスは1KBアラインされている必要がある。

(2) CSRレジスタの更新

　例外または割り込みに入ると、マイクロプロセッサは関連するCSRレジスタを自動的に更新する。これには、Machineモード例外原因レジスタmcause、Machineモード例外ポインタレジスタmepc、およびMachineモードステータスレジスタmstatusが含まれる。

・mcauseの更新

　前述のとおり、例外または割り込みに入ると、その値は現在の例外タイプまたは割り込み番号を反映する。ソフトウェアはこのレジスタ値を読み取ることで、例外の原因を確認したり、割り込みの発生源を特定したりすることができる。詳細は表2-1を参照。

・mepcの更新

例外または割り込みから復帰する際のマイクロプロセッサの戻りアドレスは、標準的にmepcに格納される。そのため、例外または割り込みが発生すると、ハードウェアはmepcの値を、例外発生時の現在の命令PC値、または割り込み前に事前実行された次の命令のPC値に自動的に更新する。例外または割り込みの処理が完了すると、マイクロプロセッサは保存されたこの値を戻りアドレスとして使用し、割り込み発生位置に復帰して処理を継続する。

ただし、注意すべき点がある。

1. mepcは読出し／書込み可能なレジスタであり、ソフトウェアによって値を変更することで、例外／割り込み復帰後に実行されるPCポインタの位置を変更することができる。
2. 割り込みが発生した場合、すなわち例外原因レジスタmcause[31]==1のとき、mepcの値は割り込み発生時点で未実行の次の命令のPC値に更新される。

例外が発生した場合、すなわち例外原因レジスタmcause[31]==0のとき、mepcの値は現在の例外に対応する命令のPC値に更新される。このため、例外から復帰する際にmepcの値をそのまま使用して戻ると、例外を引き起こした命令を再度実行することになり、再び例外に入る可能性がある。通常、例外処理後にはmepcの値を次の未実行命令のPC値にソフトウェアで修正してから復帰する。

例えば、ecall／ebreakによって例外が発生した場合、これらの命令（c.ebreakは2バイト）は4バイト命令であるため、例外処理後にmepcの値をmepc+4（c.ebreakの場合はmepc+2）に修正してから復帰すればよい。

・mstatusの更新

例外および割り込みに入ると、ハードウェアはmstatusレジスタ内の特定ビットを更新する。

1. MPIEは、例外または割り込みに入る前のMIEの値に更新され、例外または割り込み処理後にMIEを復元するために使用される。
2. MPPビットは、例外または割り込みに入る前の特権モードを保存するために使用される。QingKe V2はMachineモードのみをサポートするため、0b11のまま変更されない。
3. QingKe V2マイクロプロセッサはMachineモードにおける割り込みネストをサポートしており、最下層の例外および割り込みに入る前にMIEはクリアされない。
4. アウトスタックフラグMPOPは、現在の例外または割り込みのアウトスタックフラグに更新され、MPPOPはMPOPに更新される。

2.3 例外処理関数

例外または割り込みに入ると、マイクロプロセッサはmtvecレジスタで定義されたアドレスおよびモードに従ってプログラムを実行する。

統一エントリ方式を使用する場合、マイクロプロセッサはmtvec[31:2]で定義されたベースアドレスから、mtvec[1]の値に基づいてジャンプ命令を取得するか、例外および割り込み処理関数のエントリアドレスを取得して実行に移る。このとき、例外および割り込み処理関数は、mcause[31]の値に基づいて原因が例外か割り込みかを判定し、例外コードにより例外の種類および原因、または対応する割り込みを判断して処理を行う。

ベースアドレス＋割り込み番号×4のオフセット方式を使用する場合、ハードウェアは割り込み番号に基づいてベクタテーブルに自動的にジャンプし、例外または割り込み処理関数のエントリアドレスを取得して実行に移る。

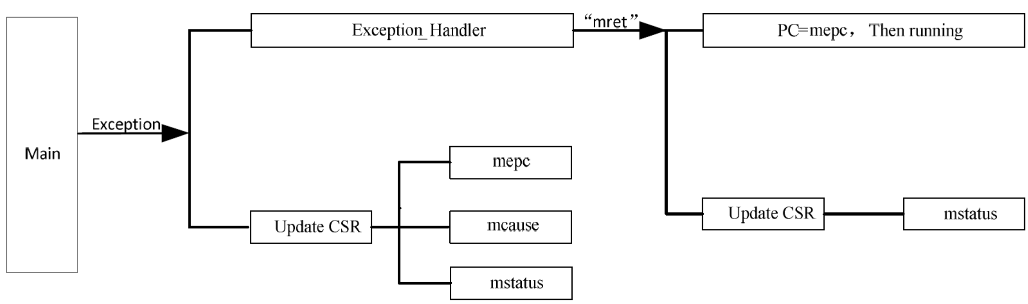
2.4 例外からの復帰

例外または割り込み処理が完了した後は、サービスプログラムからの復帰が必要である。例外および割り込みに入ると、マイクロプロセッサはUserモードからMachineモードに移行し、例外および割り込みの処理もMachineモードで完了する。例外および割り込みから復帰する際には、mret命令を使用して戻る必要がある。このとき、マイクロプロセッサのハードウェアは以下の操作を自動的に行う。

- PCポインタはCSRレジスタmepcの値に復元され、mepcに保存された命令アドレスから実行が開始される。例外処理完了後には、mepcのオフセット操作に注意する必要がある。
- CSRレジスタmstatusが更新され、MIEはMPIEに復元され、MPOPはMPPOPに復元される。

例外応答の一連の処理は、以下の図2-1によって説明される。

図2-1 例外応答処理フロー



Chapter 3 PFICおよび割り込み制御

QingKe V2マイクロプロセッサには、例外を含む最大256個の割り込みを管理可能なProgrammable Fast Interrupt Controller(PFIC)が設計されている。そのうち最初の16個はマイクロプロセッサの内部割り込みとして固定されており、残りは外部割り込みであり、最大240個まで拡張可能である。主な特徴は以下のとおりである。

- 240個の外部割り込み:各割り込み要求は独立したトリガおよびマスク制御ビットを持ち、専用ステータスビットを備える
- 割り込み優先度のプログラム設定が可能で、2段階のネストをサポート
- 高速割り込み入出機構:ハードウェアによる自動スタックおよび復帰、最大HPE深度は2段階
- Vector Table Free(VTF)割り込み応答機構:割り込みベクタアドレスへの2チャネルのプログラム可能な直接アクセスをサポート

例外および割り込みのベクタテーブルは、以下の表3-1に示される。

表3-1 例外および割り込みベクタテーブル

Number	Priority	Type	Name	Description
0	-	-	-	-
1	-	-	-	-
2	-2	Fixed	NMI	Non-maskable interrupt
3	-1	Fixed	EXC	Exception interrupt
4-11	-	-	-	-
12	0	Programmable	SysTick	System timer interrupt
13	-	-	-	-
14	1	Programmable	SWI	Software interrupt
15	-	-	-	-
16-255	2-241	Programmable	External interrupt	External interrupt 16-255

3.1 PFIC Register Set

表3-2 PFICレジスタ

Name	Access address	Access	Description	Reset value
PFIC_ISRx	0xE000E000 -0xE000E01C	RO	割り込み許可ステータスレジスタ x	0x00000000
PFIC_IPRx	0xE000E020 -0xE000E03C	RO	割り込み保留ステータスレジスタ x	0x00000000
PFIC_ITHRESDR	0xE000E040	RW	割り込み優先度しきい値設定レジスタ	0x00000000
PFIC_CFGR	0xE000E048	RW	割り込み設定レジスタ	0x00000000
PFIC_GISR	0xE000E04C	RO	割り込みグローバルステータスレジスタ	0x00000000
PFIC_VTFIDR	0xE000E050	RW	VTF ID 設定レジスタ	0x00000000
PFIC_VTFADDRRx	0xE000E060 -0xE000E06C	RW	VTF x オフセットアドレスレジスタ x	0x00000000
PFIC_IENRx	0xE000E100 -0xE000E11C	WO	割り込み有効設定レジスタ x	0x00000000
PFIC_IRERx	0xE000E180 -0xE000E19C	WO	割り込み有効クリアレジスタ x	0x00000000
PFIC_IPSRx	0xE000E200 -0xE000E21C	WO	割り込み保留設定レジスタ x	0x00000000
PFIC_IPRRx	0xE000E280 -0xE000E29C	WO	割り込み保留クリアレジスタ x	0x00000000
PFIC_IACTRx	0xE000E300 -0xE000E31C	RO	割り込みアクティベーションステータスレジスタ x	0x00000000
PFIC_IPRIORx	0xE000E400 -0xE000E43C	RW	割り込みプライオリティ設定レジスタ	0x00000000
PFIC_SCTLR	0xE000ED10	RW	システムコントロールレジスタ	0x00000000

注記：
1．PFIC_ISR0レジスタの初期値は0xCであり、これはNMIおよび例外が常にデフォルトで有効であることを意味する。
2．NMIおよびEXCは割り込み保留のクリアおよび設定操作をサポートするが、割り込み許可のクリアおよび設定操作はサポートしない。

各レジスタの説明は以下のとおりである。

割り込み許可ステータスレジスタおよび割り込み保留ステータスレジスタ（PFIC_ISR<0–7>／PFIC_IPR<0–7>）

Name	Access address	Access	Description	Reset value
PFIC_ISR0	0xE000E000	RO	割り込み0～31許可ステータスレジスタ 合計32ビットのステータスビット[n]を持ち、#n割り込みの許可状態を示す。 注記：NMIおよびEXCはデフォルトで有効になっている。	0x0000000C
PFIC_ISR1	0xE000E004	RO	割り込み32～63許可ステータスレジスタ 合計32ビットのステータスビットを持ち、各ビットは対応する割り込みの許可状態を示す。	0x00000000
…	…	…	…	…
PFIC_ISR7	0xE000E01C	RO	割り込み224～255許可ステータスレジスタ	0x00000000
PFIC_IPR0	0xE000E020	RO	割り込み0～31保留ステータスレジスタ 合計32ビットのステータスビット[n]を持ち、#n割り込みの保留状態を示す。	0x00000000
PFIC_IPR1	0xE000E024	RO	割り込み32～63保留ステータスレジスタ	0x00000000
…	…	…	…	…
PFIC_IPR7	0xE000E03C	RO	割り込み224～255保留ステータスレジスタ	0x00000000

対応する割り込みを有効化／無効化するために、2種類のレジスタが使用される。

割り込み許可設定レジスタおよび割り込み許可クリアレジスタ（PFIC_IENR<0–7>／PFIC_IRER<0–7>）

Name	Access address	Access	Description	Reset value
PFIC_IENR0	0xE000E100	WO	割り込み0～31許可設定レジスタ 合計32ビットの設定ビット[n]を持ち、#n割り込みの許可設定に使用される。 注記：NMIおよびEXCはデフォルトで有効になっている。	0x00000000
PFIC_IENR1	0xE000E104	WO	割り込み32～63許可設定レジスタ	0x00000000
…	…	…	…	…
PFIC_IENR7	0xE000E11C	WO	割り込み224～255許可設定レジスタ	0x00000000
-	-	-	-	-
PFIC_IRER0	0xE000E180	WO	割り込み0～31許可クリアレジスタ 合計32ビットのクリアビット[n]を持ち、#n割り込みの許可クリアに使用される。 注記：NMIおよびEXCは操作できない。	0x00000000
PFIC_IRER1	0xE000E184	WO	割り込み32～63許可クリアレジスタ	0x00000000
…	…	…	…	…
PFIC_IRER7	0xE000E19C	WO	割り込み224～255許可クリアレジスタ	0x00000000

対応する割り込みを有効化／無効化するために、2種類のレジスタが使用される。
注記：任意の割り込みをマスクするためにレジスタを使用する場合、またはCSRレジスタを使用してグローバル割り込みをマスクする場合は、コア制御状態と割り込み許可状態の同期を取るために 'fence.i' 命令を追加すること。

割り込み保留設定レジスタおよび割り込み保留クリアレジスタ（PFIC_IPSR<0–7>／PFIC_IPRR<0–7>）

Name	Access address	Access	Description	Reset value
PFIC_IPSR0	0xE000E200	WO	割り込み0～31保留設定レジスタ 32ビットの設定ビット[n]を持ち、#n割り込みの保留設定に使用される。	0x00000000
PFIC_IPSR1	0xE000E204	WO	割り込み32～63保留設定レジスタ	0x00000000
…	…	…	…	…
PFIC_IPSR7	0xE000E21C	WO	割り込み224～255保留設定レジスタ	0x00000000
-	-	-	-	-
PFIC_IPRR0	0xE000E280	WO	割り込み0～31保留クリアレジスタ 合計32ビットのクリアビット[n]を持ち、#n割り込みの保留クリアに使用される。	0x00000000
PFIC_IPRR1	0xE000E284	WO	割り込み32～63保留クリアレジスタ	0x00000000
…	…	…	…	…
PFIC_IPRR7	0xE000E29C	WO	割り込み224～255保留クリアレジスタ	0x00000000

マイクロプロセッサが割り込みを有効化すると、割り込み保留レジスタを介して直接割り込みを設定し、割り込みをトリガすることができる。割り込み保留クリアレジスタを使用して、保留トリガをクリアする。

割り込みアクティベーションステータスレジスタ（PFIC_IACTR<0–7>）

Name	Access address	Access	Description	Reset value
PFIC_IACTR0	0xE000E300	RO	割り込み0～31アクティベーションステータスレジスタ 32ビットのステータスビット[n]を持ち、#n割り込みが実行中であることを示す。	0x00000000
PFIC_IACTR1	0xE000E304	RO	割り込み32～63アクティベーションステータスレジスタ	0x00000000
…	…	…	…	…
PFIC_IACTR7	0xE000E31C	RO	割り込み224～255アクティベーションステータスレジスタ	0x00000000

各割り込みにはアクティブステータスビットがあり、割り込みに入ると設定され、`mret` によってリターンする際にハードウェアによってクリアされる。

割り込み優先度レジスタおよび優先度しきい値レジスタ(PFIC_IPRIOR<0–7>／PFIC_ITHRESDR)

Name	Access address	Access	Description	Reset value
PFIC_IPRIOR0	0xE000E400	RW	割り込み0優先度設定 [7:6]:優先度制御ビット - ネスト構成でない場合、プリエンプションビットは存在しない - ネスト構成の場合、bit7がプリエンプションビットとなる [5:0]:予約済み、0に固定 注記:優先度値が小さいほど、優先度は高くなる。同一のプリエンプション優先度で複数の割り込みが同時に保留された場合、より高い優先度の割り込みが先に実行される。	0x00
PFIC_IPRIOR1	0xE000E401	RW	割り込み1優先度設定	0x00
PFIC_IPRIOR2	0xE000E402	RW	割り込み2優先度設定	0x00
…	…	…	…	…
PFIC_IPRIOR254	0xE000E4FE	RW	割り込み254優先度設定	0x00
PFIC_IPRIOR255	0xE000E4FF	RW	割り込み255優先度設定	0x00
-	-	-	-	-
PFIC_ITHRESDR	0xE000E040	RW	割り込み優先度しきい値設定 [31:8]:予約済み、0に固定 [7:6]:優先度しきい値 [5:0]:予約済み、0に固定 注記:優先度値がしきい値以上の割り込みについては、保留状態が発生しても割り込みサービス関数は実行されない。このレジスタが0の場合、しきい値レジスタは無効であることを意味する。	0x00

割り込み設定レジスタ (PFIC_CFGR)

Name	Access address	Access	Description	Reset value
PFIC_CFGR	0xE000E048	RW	割り込み設定レジスタ	0x00000000

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE	WO	対応するターゲット制御ビットに対しては、対応するセキュリティアクセス識別データを同時に書き込む必要があり、読出しデータは0に固定される。 KEY1 = 0xFA05 KEY2 = 0xBCAF KEY3 = 0xBEEF	0
[15:8]	Reserved	RO	Reserved	0
7	SYSRESET	WO	システムリセット(KEY3を同時に書き込む) 自動で0にクリアされる 1の書き込みは有効、0の書き込みは無効 注記:PFIC_SCTLRレジスタのSYSRESETビットと同一の機能を持つ。	0
[6:0]	Reserved	RO	Reserved	0

V2シリーズマイクロプロセッサ
このレジスタは主に互換性のために使用される。

割り込みグローバルステータスレジスタ(PFIC_GISR)

Name	Access address	Access	Description	Reset value
PFIC_GISR	0xE000E04C	RO	割り込みグローバルステータスレジスタ	0x00000000

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
9	GPENDSTA	RO	1: 割り込みが現在保留中である 0: 保留中ではない	0
8	GACTSTA	RO	1: 割り込み処理中である 0: 割り込み処理中ではない	0
[7:0]	NESTSTA	RO	現在の割り込みネスト状態 0x03:レベル2割り込み中 0x01:レベル1割り込み中 0x00:割り込み未発生 その他:あり得ない状態	0

VTF IDとアドレスレジスタ(PFIC_VTFIDR/PFIC_VTFADDRR<0-1>)

Name	Access address	Access	Description	Reset value
PFIC_VTFIDR	0xE000E050	RW	[15:8]: number of VTF 1 [7:0]: number of VTF 0	0x00000000
-	-	-	-	-
PFIC_VTFADDRR0	0xE000E060	RW	[31:1]: VTF 0 address, two-byte alignment [0]: 1: Enable VTF 0 channel 0: Close	0x00000000
PFIC_VTFADDRR1	0xE000E064	RW	[31:1]: VTF 1 address, two-byte alignment [0]: 1: Enable VTF 1 channel 0: Close	0x00000000

システムコントロールレジスタ (PFIC_SCTLR)

Name	Access address	Access	Description	Reset value
PFIC_SCTLR	0xE000ED10	RW	System control register	0x00000000

Bit	Name	Access	Description	Reset value
31	SYSRESET	WO	システムリセット、自動的に0にクリアされる。 1の書き込みは有効 0の書き込みは無効 PFIC_CFGRレジスタと同一の効果を持つ。	0
[30:6]	Reserved	RO	Reserved	0
5	SETEVENT	WO	WFE(Wait For Event)でウェイクアップするためのイベントを設定する。	0
4	SEVONPEND	RW	イベントが発生するか、割り込みが保留状態にある場合、WFE命令の実行後にシステムはウェイクアップされる。また、WFE命令が実行されていない場合は、次の命令実行直後にシステムは即座にウェイクアップされる。 1:有効なイベントおよびすべての割り込み(無効化された割り込みを含む)でシステムをウェイクアップ可能 0:有効なイベントおよび有効な割り込みのみでシステムをウェイクアップ可能	0
3	WFITOWFE	RW	WFI命令をWFE命令として実行する。 1:以降のWFI命令をWFE命令として扱う 0:効果なし	0
2	SLEEPDEEP	RW	制御システムの低電力モード 1:ディープスリープ 0:スリープ	0
1	SLEEPONEXIT	RW	割り込みサービスプログラムから制御が離れた後のシステム状態 1:システムは低電力モードに入る 0:システムはメインプログラムにもどる	0
0	Reserved	RO	Reserved	0

3.2 Interrupt-related CSR Registers

さらに、以下のCSRレジスタも割り込み処理に大きな影響を与える。

Interrupt System Control Register (INTSYSCR)

Name	CSR Address	Access	Description	Reset value
INTSYSCR	0x804	MRW	Interrupt system control register	0x00000000

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	MRO	Reserved	0
2	EABIEN	MRW	EABI enable. 0: EABI off. 1: EABI enabled.	0
1	INESTEN	MRW	Interrupt nesting enable. 0: Interrupt nesting function off. 1: Interrupt nesting function enabled.	0
0	HWSTKEN	MRW	HPE enable. 0: HPE function off. 1: HPE function enabled.	0

Machine Mode Exception Base Address Register (mtvec)

Name	CSR Address	Access	Description	Reset value
mtvec	0x305	MRW	Exception base address register	0x00000000

Bit	Name	Access	Description	Reset value
[31:2]	BASEADDR[31:2]	MRW	割り込みベクタテーブルのベースアドレス。1KBアラインされている必要がある。	0
1	MODE1	MRW	割り込みベクタテーブルの識別方式 0:ジャンプ命令による識別。範囲は限定されるが、非ジャンプ命令もサポート可能 1:絶対アドレスによる識別。全範囲をサポートするが、ジャンプが必須となる	0
0	MODE0	MRW	割り込みまたは例外のエントリアドレスモード選択 0:共通のエントリアドレスを使用 1:割り込み番号 × 4 に基づくアドレスオフセットを使用	0

V2シリーズマイクロプロセッサのMCUでは、スタートアップファイルにおいて `MODE[1:0] == 11` がデフォルトで設定されている。すなわち、ベクタテーブルは割り込み関数の絶対アドレスを使用し、例外または割り込みのエントリは「割り込み番号 × 4 」に基づいてオフセットされる。

3.3 割り込みのネスト

割り込みネストは、割り込みシステム制御レジスタ INTSYSCR（CSRアドレス：0x804）および割り込み優先度レジスタ PFIC_IPRIOR を併用することで許可される。V2シリーズMCUのスタートアップファイルにて、割り込みシステム制御レジスタでネストを有効化し、対応する割り込みの優先度を設定する。

優先度値が小さいほど、優先度は高くなる。プリエンプションビットの値が小さいほど、プリエンプション優先度は高くなる。同一のプリエンプション優先度で複数の割り込みが同時に保留された場合、マイクロプロセッサはより優先度の値が小さい（優先度の高い）割り込みに先に応答する。

3.4 ハードウェアプロローグ／エピローグ（HPE）

例外または割り込みが発生すると、マイクロプロセッサは現在のプログラムフローを停止し、例外または割り込み処理関数の実行に移行する。この際、現在のプログラムフローの「位置」を保存する必要がある。例外または割り込みから復帰した後は、その「位置」を復元し、停止していたプログラムフローの実行を継続する必要がある。

V2シリーズマイクロプロセッサにおいて、この「位置」とは、表1-2 に示される Caller Saved レジスタ全体を指す。V2シリーズでは、ハードウェアがこれらの整形済み Caller Saved レジスタのうち10個をユーザスタック領域に自動的に保存し、例外または割り込みから復帰する際に、スタック領域から10個の整形レジスタへ自動的にデータを復元する。

HPEはネストに対応しており、最大ネスト深度は2レベルである。スタック開始時点のSP値をNとすると、スタック挿入時にSP値は48バイト分オフセットされ、SPはN−48に更新される。スタック挿入後のスタック内部の変化は、以下の表に示される。

表3-3 V2プロセッサ スタック挿入順序

SP Address	Register	Description
Old SP value(N-0)	-	The contents of the stack already pressed in
N-4	x1	Automatic saving of x1 values
N-8	x5	Automatic saving of x5 values
N-12	x6	Automatic saving of x6 values
N-16	x7	Automatic saving of x7 values
N-20	x10	Automatic saving of x10 values
N-24	x11	Automatic saving of x11 values
N-28	x12	Automatic saving of x12 values
N-32	x13	Automatic saving of x13 values
N-36	x14	Automatic saving of x14 values
N-40	x15	Automatic saving of x15 values
…	…	…
New SP value(N-48)	-	SP value after entering the stack

注記：
1．HPE（Hardware Prologue/Epilogue）を使用する割り込み関数は、MRSまたはその提供するツールチェーンでコンパイルする必要があり、割り込み関数は `__attribute__((interrupt("WCH-Interrupt-fast")))` によって宣言されなければならない。
2．スタックプッシュを使用する割り込み関数は、`__attribute__((interrupt()))` によって宣言される。

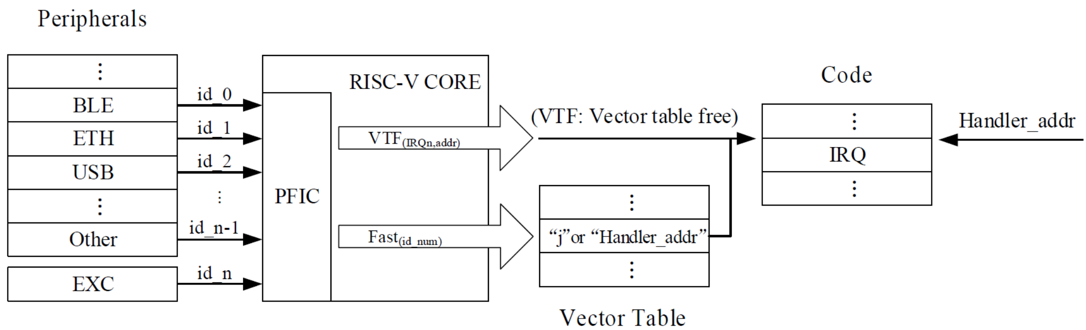
3.5 ベクタテーブルフリー（VTF）

プログラマブル高速割り込みコントローラ（PFIC）は、2つのVTFチャネルを提供しており、割り込みベクタテーブルの検索処理を介さずに、割り込み関数のエントリに直接アクセスできる。

通常の割り込み関数を設定する際には、割り込み番号をVTF IDレジスタ（PFIC_VTFIDR）のチャネルxに書き込み、対応するVTFアドレスレジスタ（VTFADDRRx）に割り込み関数のエントリアドレスを書き込み、そのチャネルのVTFを有効化する。

PFICは、高速割り込みおよびVTFに対して、図3-2 に示されるように応答する。

図3-2 プログラマブル高速割り込みコントローラ（PFIC）



3.6 テールチェイニングおよびレイトアライバル

QingKe V2シリーズのHPEは、レジスタをユーザスタック領域に保存し、優先度設定および割り込みネストをサポートする。テールチェイニング現象とは、高優先度割り込みに応答している際に、同時に低優先度割り込みが保留されている場合を指す。高優先度割り込みの実行後、高優先度割り込みの中間終了時のスタックアウト処理を省略し、保留割り込みの実行前にスタックイン処理を行うことで、割り込みレイテンシを低減する。

同様に、プロセッサが低優先度割り込みのスタックイン処理中に、高優先度割り込み要求を遅れて受信した場合、高優先度割り込みはスタックイン処理を繰り返さず、直前の処理に直接続行する。

Chapter 4 システムタイマ（SysTick）

　QingKe V2シリーズマイクロプロセッサには、32ビット加算カウンタ（SysTick）が内蔵されており、そのクロックソースはシステムクロックまたはシステムクロックの8分周を選択可能である。リアルタイムオペレーティングシステムの時間ベースの提供、タイミング、時間計測などに使用できる。タイマは4つのレジスタを含み、SysTickの制御のためにペリフェラルアドレス空間にマッピングされている。詳細を 表4-1 に示す。

表4-1 SysTick register list

Name	Access address	Description	Reset value
STK_CTLR	0xE000F000	System count control register	0x00000000
STK_SR	0xE000F004	System count status register	0x00000000
STK_CNTR	0xE000F008	System counter register	0x00000000
STK_CMPR	0xE000F010	System count comparison value register	0x00000000

System Count Control Register (STK_CTLR)

表4-2 SysTick Control Registers

Bit	Name	Access	Description	Reset value
31	SWIE	RW	ソフトウェア割り込みトリガ有効化(SWI) 1:ソフトウェア割り込みをトリガする 0:トリガを無効化する ソフトウェア割り込みに入った後は、ソフトウェアによるクリア(0)が必要。クリアしない場合、継続的にトリガされる。	0
[30:4]	Reserved	RO	Reserved	0
3	STRE	RW	オートリロードカウント有効ビット 1:比較値までカウントし、0から再カウントを開始する 0:カウントアップを継続する	0
2	STCLK	RW	カウンタクロックソース選択ビット 1:時間ベースとして HCLK を使用 0:時間ベースとして HCLK/8 を使用	0
1	STIE	RW	カウンタ割り込み有効制御ビット 1:カウンタ割り込みを有効化する 0:カウンタ割り込みを無効化する	0
0	STE	RW	システムカウンタ有効制御ビット 1:システムカウンタ STK を開始する 0:システムカウンタ STK を停止し、カウンタはカウントを停止する	0

System Count Status Register (STK_SR)

表4-3 SysTick Status Register

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	CNTIF	RW0	カウント値比較フラグ(0書き込みでクリア、1書き込みは無効) 1:カウントアップにより比較値に到達した 0:比較値に到達していない	0

System Counter Value Register (STK_CNTR)

Bit	Name	Access	Description	Reset value
[31:0]	CNTR	RW	Current counter count value.	0

System Count Comparison Value Register (STK_CMPR)

Bit	Name	Access	Description	Reset value
[31:0]	CMPR	RW	Set counter comparison value	0

Chapter 5 プロセッサ低消費電力設定

QingKe V2シリーズマイクロプロセッサは、WFI(Wait For Interrupt)命令によりスリープ状態をサポートし、低静的消費電力を実現する。PFICのシステム制御レジスタ(PFIC_SCTLR)と組み合わせることで、各種スリープモードおよびWFEコマンドの実行が可能である。

5.1 スリープへの移行

QingKe V2シリーズマイクロプロセッサは、Wait for Interrupt(WFI)およびWait For Event(WFE)の2つの方法でスリープ状態に移行可能である。WFI方式は、マイクロプロセッサがスリープ状態に入り、割り込みによって起床し、対応する割り込みを実行することを意味する。WFE方式は、マイクロプロセッサがスリープ状態に入り、イベントによって起床し、停止していたプログラムフローを継続して実行することを意味する。

標準RISC-VはWFI命令をサポートしており、WFIコマンドを直接実行することでWFI方式によるスリープに入ることができる。WFE方式を使用するには、システム制御レジスタPFIC_SCTLR の WFITOWFE ビットを使用して、以降のWFIコマンドをWFE処理として制御し、WFE方式によるスリープを実現する。

スリープの深度は PFIC_SCTLR の SLEEPDEEP ビットによって制御される。

- PFIC_SCTLRレジスタのSLEEPDEEPが0にクリアされている場合、マイクロプロセッサはスリープモードに入り、SysTickおよび一部のウェイクアップロジックを除いて内部ユニットクロックの停止が許可される。
- PFIC_SCTLRレジスタのSLEEPDEEPがセットされている場合、マイクロプロセッサはディープスリープモードに入り、すべてのセルクロックの停止が許可される。

マイクロプロセッサがデバッグモードにある場合、いかなるスリープモードにも入ることはできない。

5.2 スリープからのウェイクアップ

　QingKe V2シリーズマイクロプロセッサは、WFIおよびWFEによるスリープ後、以下の方法でウェイクアップ可能である。

・WFI方式によるスリープ後のウェイクアップ:

1. 割り込みコントローラが応答する割り込みソースによってマイクロプロセッサがウェイクアップされる。ウェイクアップ後、マイクロプロセッサは割り込み関数を最初に実行する。
2. スリープモードに入った場合、デバッグ要求によってマイクロプロセッサをウェイクアップさせることができる。ディープスリープに入った場合、デバッグ要求ではマイクロプロセッサをウェイクアップできない。

・WFE方式によるスリープ後、マイクロプロセッサは以下の方法でウェイクアップ可能である。

1. 内部または外部イベントにより、割り込みコントローラの設定を必要とせずにウェイクアップし、プログラムの実行を継続する。
2. 割り込みソースが有効化されている場合、割り込みが発生するとマイクロプロセッサはウェイクアップし、割り込み関数を最初に実行する。
3. PFIC_SCTLR の SEVONPEND ビットが設定されている場合、割り込みコントローラが割り込みを有効化していなくても、新たな割り込み保留信号が生成されると（以前に生成された保留信号は無効）、マイクロプロセッサをウェイクアップさせることができる。ウェイクアップ後、対応する割り込み保留フラグは手動でクリアする必要がある。
4. スリープモードに入った場合、デバッグ要求によりマイクロプロセッサをウェイクアップさせることができる。ディープスリープに入った場合、デバッグ要求ではマイクロプロセッサをウェイクアップできない。

　また、ウェイクアップ後のマイクロプロセッサの状態は、PFIC_SCTLR の SLEEPONEXIT ビットを設定することで制御可能である。

・SLEEPONEXIT が設定されている場合、最下層の割り込み復帰命令（mret）により WFI モードのスリープがトリガされる。

・SLEEPONEXIT がクリアされている場合、効果はない。

　V2シリーズマイクロプロセッサを搭載した各種MCU製品は、PFIC_SCTLRの設定に応じて異なるスリープモードを採用し、異なるペリフェラルおよびクロックを停止し、異なる電源管理ポリシーおよびウェイクアップ方式を実装することで、各種低消費電力モードを実現可能である。

Chapter 6 デバッグサポート

QingKe V2シリーズマイクロプロセッサには、複雑なデバッグ操作をサポートするハードウェアデバッグモジュールが含まれている。マイクロプロセッサがサスペンド状態のとき、デバッグモジュールは抽象コマンド、プログラムバッファ展開命令などを通じて、マイクロプロセッサのGPR（汎用レジスタ）、CSR（制御ステータスレジスタ）、メモリ、外部デバイスなどへアクセス可能である。デバッグモジュールは、マイクロプロセッサの動作をサスペンドおよびレジュームすることができる。

デバッグモジュールは、RISC-V External Debug Support Version 0.13.2 仕様に準拠しており、詳細なドキュメントは RISC-V International のウェブサイトからダウンロード可能である。

6.1 デバッグモジュール

マイクロプロセッサ内のデバッグモジュールは、デバッグホストから発行されるデバッグ操作を実行可能であり、以下を含む：

- デバッグインタフェースを介したレジスタへのアクセス
- デバッグインタフェースを介したマイクロプロセッサのリセット、サスペンドおよびレジューム
- デバッグインタフェースを介したメモリ、命令レジスタおよび外部デバイスの読み書き
- デバッグインタフェースを介した任意命令の複数展開
- デバッグインタフェースを介したソフトウェアブレークポイントの設定
- シングルステップデバッグのサポート

デバッグモジュールの内部レジスタは7ビットのアドレスコードを使用し、以下のレジスタが QingKe V2シリーズマイクロプロセッサ内に実装されている。

表6-1 Debug module register List

Name	Access address	Description
data0	0x04	データレジスタ0、一時的なデータ保存に使用可能
data1	0x05	データレジスタ1、一時的なデータ保存に使用可能
dmcontrol	0x10	Debug module control register
dmstatus	0x11	Debug module status register
hartinfo	0x12	Microprocessor status register
abstractcs	0x16	Abstract command status register
command	0x17	Abstract command register
progbuf0-7	0x20-0x27	Instruction cache registers 0-7
haltsum0	0x40	Pause status register

デバッグホストは、dmcontrolレジスタを設定することで、マイクロプロセッサのサスペンド、レジューム、リセットなどを制御可能である。RISC-V標準では、抽象コマンドとして「レジスタアクセス」「高速アクセス」「メモリアクセス」の3種類が定義されている。QingKe V2マイクロプロセッサは、抽象コマンドを通じてレジスタ(GPRs、CSRs、FPRs)へのアクセスをサポートする。

デバッグモジュールは、8個の命令キャッシュレジスタ progbuf0～progbuf7 を実装しており、デバッグホストは複数の命令(圧縮命令も含む)をバッファにキャッシュ可能である。抽象コマンドの実行後に命令キャッシュレジスタ内の命令を継続して実行するか、キャッシュされた命令を直接実行するかを選択できる。なお、progbufs内の命令が32バイト未満の場合、最後の命令は "ebreak" または "c.ebreak" である必要があり、命令が32バイトに達する場合、デバッグモジュールは自動的に "ebreak" 命令を追加する。

デバッグホストは、抽象コマンドおよび progbufs にキャッシュされた命令、さらにストレージ、ペリフェラルなどにもアクセス可能である。

各レジスタの詳細を以下に記載する。

Data Register 0 (data0)

表6-2 data0 register definition

Bit	Name	Access	Description	Reset value
[31:0]	data0	RW	Data register 0, used for temporary storage of data	0

Data Register 1 (data1)

表6-3 data1 register definition

Bit	Name	Access	Description	Reset value
[31:0]	data1	RW	Data register 1, used for temporary storage of data	0

Debug Module Control Register (dmcontrol)

このレジスタは、マイクロプロセッサの一時停止、リセット、および再開を制御する。デバッグホストは、対応するフィールドにデータを書き込むことで、一時停止(haltreq)、リセット(ndmreset)、再開(resumereq)を実行する。以下に詳細を記述する。

表6-4 dmcontrol register definition

Bit	Name	Access	Description	Reset value
31	haltreq	WIO	0: Clear the pause request 1: Send a pause request	0
30	resumereq	W1	0: Invalid 1: Restore the current microprocessor 注:1の書き込みが有効であり、マイクロプロセッサの復元後にハードウェア側でクリアされる	0
29	Reserved	RO	Reserved	0
28	ackhavereset	W1	0: Invalid 1: Clear the haverest status bit of the microprocessor	0
[27:2]	Reserved	RO	Reserved	0
1	ndmreset	RW	0: Clear reset 1: Reset the entire system other than the debug module	0
0	dmactive	RW	0: Reset debug module 1: Debug module works properly	0

Debug Module Status Register (dmstatus)

　このレジスタは、デバッグモジュールのステータスを示すために使用される読み取り専用レジスタであり、各ビットの説明は以下のとおりである。

表6-5 dmstatus register definition

Bit	Name	Access	Description	Reset value
[31:20]	Reserved	RO	Reserved	0
19	allhavereset	RO	0: Invalid 1: Microprocessor reset	0
18	anyhavereset	RO	0: Invalid 1: Microprocessor reset	0
17	allresumeack	RO	0: Invalid 1: Microprocessor reset	0
16	anyresumeack	RO	0: Invalid 1: Microprocessor reset	0
[15:14]	Reserved	RO	Reserved	0
13	allavail	RO	0: Invalid 1: Microprocessor is not available	0
12	anyavail	RO	0: Invalid 1: Microprocessor is not available	0
11	allrunning	RO	0: Invalid 1: Microprocessor is running	0
10	anyrunning	RO	0: Invalid 1: Microprocessor is running	0
9	allhalted	RO	0: Invalid 1: Microprocessor is in suspension	0
8	anyhalted	RO	0: Invalid 1: Microprocessor out of suspension	0
7	authenticated	RO	0: Authentication is required before using the debug module 1: The debugging module has been certified	0x1
[6:4]	Reserved	RO	Reserved	0
[3:0]	version	RO	Debugging system support architecture version 0010: V0.13	0x2

Microprocessor Status Register (hartinfo)

このレジスタは、マイクロプロセッサに関する情報をデバッグホストに提供するために使用される読み取り専用レジスタであり、各ビットの説明は以下のとおりである。

表6-6 hartinfo register definition

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:20]	nscratch	RO	Number of dscratch registers supported	0x2
[19:17]	Reserved	RO	Reserved	0
16	dataaccess	RO	0: Data register is mapped to CSR address 1: Data register is mapped to memory address	0x1
[15:12]	datasize	RO	Number of data registers	0x2
[11:0]	dataaddr	RO	Data register data0 offset address, the base address is 0xe0000000	0x0f4

Abstract Command Control and Status Registers (abstractcs)

　このレジスタは、抽象コマンドの実行状況を示すために使用される。デバッグホストはこのレジスタを読み取ることで、直前の抽象コマンドが実行されたかどうかを確認でき、また実行中にエラーが発生したかどうか、およびそのエラーの種類を確認することができる。詳細は以下に記述される。

表 6-7 abstractcs register definitions

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
[28:24]	progbufsize	RO	Indicates the number of program buffer program cache registers	0x8
[23:13]	Reserved	RO	Reserved	0
12	busy	RO	0: No abstract command is executing 1: There are abstract commands being executed Note: After execution, the hardware is cleared.	0
11	Reserved	RO	Reserved	0
[10:8]	cmder	RW	Abstract command error type 000: No error 001: abstract command execution to write to command, abstractcs, abstractauto registers or read and write to data and progbuf registers 010: Does not support current abstract command 011: Execution of abstract command with exception 100: The microprocessor is not suspended or unavailable and cannot execute abstract commands 101: Bus error 110: Parity bit error during communication 111: Other errors Note: For bit writing 1 is used to clear the zero.	0
[7:4]	Reserved	RO	Reserved	0
[3:0]	datacount	RO	Number of data registers	0x2

Abstract Command Register(command)

　デバッグホストは、抽象コマンドレジスタに異なる設定値を書き込むことで、マイクロプロセッサ内部の GPRs、FPRs、CSRs レジスタにアクセス可能である。

　レジスタアクセス時のコマンドレジスタの各ビット定義は以下のとおりである。

表6-8 Definition of command register when accessing registers

Bit	Name	Access	Description	Reset value
[31:24]	cmdtype	WO	Abstract command type 0: Access register 1: Quick access (not supported) 2: Access to memory (not supported)	0
23	Reserved	WO	Reserved	0
[22:20]	aarsize	WO	Access register data bit width 000: 8-bit 001: 16-bit 010: 32-bit 011: 64-bit (not supported) 100: 128-bit (not supported) Note: When accessing floating-point registers FPRs, only 32-bit access is supported.	0
19	aarpostincrement	WO	0: No effect 1: Automatically increase the value of regno after accessing the register	0
18	postexec	WO	0: No effect 1:Execute the abstract command and then execute the command in progbuf	0
17	transfer	WO	0: Do not execute the operation specified by write 1: Execute the manipulation specified by write	0
16	write	WO	0: Copy data from the specified register to data0 1: Copy data from data0 register to the specified register	0
[15:0]	regno	WO	Specify access registers 0x0000-0x0fff are CSRs 0x1000-0x101f are GPRs 0x1020-0x103f are FPRs	0

Instruction Cache Register (progbufx)

このレジスタは任意の命令を格納し、対応する操作を展開するために使用される。8個のレジスタが含まれており、最後の実行命令は "ebreak" または "c.ebreak" である必要がある点に注意すること。

表6-9 progbuf register definition

Bit	Name	Access	Description	Reset value
[31:0]	progbuf	RW	Instruction encoding for cache operations, which may include compression instructions	0

Pause Status Register (haltsum0)

このレジスタは、マイクロプロセッサがサスペンド状態かどうかを示すために使用される。各ビットはマイクロプロセッサのサスペンド状態を示しており、コアが1つしか存在しない場合は、最下位ビットのみがその状態を示すために使用される。

表6-10 haltsum0 register definition

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	haltsum0	RO	0: Microprocessor operates normally 1: Microprocessor stop	0

上記のデバッグモジュールに関連するレジスタに加えて、デバッグ機能にはいくつかのCSRレジスタも関与しており、主にデバッグ制御およびステータスレジスタ（dcsr）とデバッグ命令ポインタ（dpc）が含まれる。これらのレジスタの詳細は以下に記述される。

Debug Control and Status Register (dcsr)

表6-11 dcsr register definition

Bit	Name	Access	Description	Reset value
[31:28]	xdebugver	DRO	0000: External debugging is not supported 0100: Support standard external debugging 1111: External debugging is supported, but does not meet the specification	0x4
[27:16]	Reserved	DRO	Reserved	0
15	ebreakm	DRW	0: The ebreak command in machine mode behaves as described in the privilege file 1: The ebreak command in machine mode can enter debug mode	0
[14:13]	Reserved	DRO	Reserved	0
12	ebreaku	DRW	0: The ebreak command in user mode behaves as described in the privilege file 1: The ebreak command in user mode can enter debug mode	0
11	stepie	DRW	0: Interrupts are disabled under single-step debugging 1: Enable interrupts under single-step debugging	0
10	Reserved	DRO	Reserved	0
9	stoptime	DRW	0: System timer running in Debug mode 1: System timer stop in Debug mode	0
[8:6]	cause	DRO	Reasons for entering debugging 001: Entering debugging in the form of ebreak command (priority 3) 010: Entering debugging in the form of trigger module (priority 4, the highest) 011: Entering debugging in the form of pause request (priority 1) 100: debugging in the form of single-step debugging (priority 0, the lowest) 101: enter debug mode directly after microprocessor reset (priority 2) Others: Reserved	0
[5:3]	Reserved	DRO	Reserved	0
2	step	DRW	0: Turn off single-step debugging 1: Enable single-step debugging	0
[1:0]	prv	DRW	Privilege mode 00: User mode 01: Supervisor mode (not supported) 10: Reserved 11: Machine mode 注:デバッグモードに入る際の特権モードを記録し、デバッガはこの値を変更することで、デバッグモードからの復帰時の特権モードを変更可能である。	0

Debug Mode Program Pointer (dpc)

このレジスタは、マイクロプロセッサがデバッグモードに入った後に実行される次の命令のアドレスを格納するために使用される。デバッグモードに入る理由に応じて、このレジスタの値は異なるルールで更新される。dpcレジスタの詳細は以下に記述される。

表6-12 dpc register definitions

Bit	Name	Access	Description	Reset value
[31:0]	dpc	DRW	Command Address	0

レジスタの更新ルールは以下の表に示される。

表6-13 dpc update rules

Enter the debugging method	dpc Update rules
ebreak	Address of the Ebreak instruction
single step	Instruction address of the next instruction of the current instruction
trigger module	Temporarily not supported
halt request	Address of the next instruction to be executed when entering Debug

6.2 デバッグインタフェース

標準の RISC-V で定義された JTAG インタフェースとは異なり、QingKe V2 シリーズのマイクロプロセッサは 1 線または 2 線のデバッグインタフェースを使用する。V2A は 1 線インタフェースのみをサポートしており、WCH デバッグインタフェースプロトコルに準拠している。

このデバッグインタフェースは、デバッグホストとデバッグモジュール間の通信を担い、デバッグホストによるデバッグモジュールレジスタの読み書き操作を実現する。

WCH は WCH_Link を設計し、その回路図およびプログラムバイナリファイルをオープンソース化しており、RISC-V アーキテクチャのすべてのマイクロプロセッサのデバッグに使用可能である。

詳細なデバッグインタフェースプロトコルについては、WCH Debug Protocol Manual を参照すること。

Chapter 7 CSR Register List

RISC-Vアーキテクチャでは、マイクロプロセッサの動作状態を制御・記録するために多数の制御およびステータスレジスタ（CSR）が定義されている。前節では一部のCSRについて紹介したが、本章では QingKe V2 シリーズマイクロプロセッサに実装されているCSRレジスタについて詳細に説明する。

7.1 CSR Register List

表7-1 List of Microprocessor CSR Registers

Type	Name	CSR Address	Access	Description
RISC-V Standard CSR	marchid	0xF12	MRO	Architecture number register
	mimpid	0xF13	MRO	Hardware implementation numbering register
	mstatus	0x300	MRW	Status register
	misa	0x301	MRW	Hardware instruction set register
	mtvec	0x305	MRW	Exception base address register
	mscratch	0x340	MRW	Machine mode staging register
	mepc	0x341	MRW	Exception program pointer register
	mcause	0x342	MRW	Exception cause register
	dcsr	0x7B0	DRW	Debug control and status registers
	dpc	0x7B1	DRW	Debug mode program pointer register
	dscratch0	0x7B2	DRW	Debug mode staging register 0
	dscratch1	0x7B3	DRW	Debug mode staging register 1
Vendor-defined CSRs	intsyscr	0x804	URW	Interrupt system control register

7.2 RISC-V Standard CSR Registers

Architecture Number Register (marchid)

このレジスタは、現在のマイクロプロセッサのハードウェアアーキテクチャ番号を示す読み取り専用レジスタであり、主にベンダコード、アーキテクチャコード、シリーズコード、およびバージョンコードで構成される。それぞれの定義は以下のとおりである。

表7-2 marchid register definition

Bit	Name	Access	Description	Reset value
31	Reserved	MRO	Reserved	1
[30:26]	Vender0	MRO	Manufacturer code 0 Fixed to the letter "W" code	0x17
[25:21]	Vender1	MRO	Manufacturer code1 Fixed to the letter "C" code	0x03
[20:16]	Vender2	MRO	Manufacturer code 2 Fixed to the letter "H" code	0x08
15	Reserved	MRO	Reserved	1
[14:10]	Arch	MRO	Architecture Code RISC-V architecture is fixed to the letter "V" code	0x16
[9:5]	Serial	MRO	Series Code QingKe V2 series, fixed to the number "4"	0x04
[4:0]	Verision	MRO	Version Code Can be the version "A", "B", "C", "F" and other letters of the code	x

製造元番号およびバージョン番号は英字で表され、シリーズ番号は数値で表される。文字の符号化表は以下の表に示される。

Table 7-3 Alphabetic Mapping Table

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

例えば、QingKe V2A マイクロプロセッサでは、このレジスタの値を読み取ると `0xDC68D841` となり、これは `WCH-V2A` に対応する。

Hardware Implementation Numbering Register (mimpid)

　このレジスタは主にベンダコードで構成されており、それぞれのコード定義は以下のとおりである。

表7-4 mimpid register definition

Bit	Name	Access	Description	Reset value
31	Reserved	MRO	Reserved	1
[30:26]	Vender0	MRO	Manufacturer code 0 Fixed to the letter "W" code	0x17
[25:21]	Vender1	MRO	Manufacturer code1 Fixed to the letter "C" code	0x03
[20:16]	Vender2	MRO	Manufacturer code 2 Fixed to the letter "H" code	0x08
15	Reserved	MRO	Reserved	1
[14:1]	Reserved	MRO	Reserved	0
0	Reserved	MRO	Reserved	1

Machine Mode Status Register (mstatus)

このレジスタは前節で一部説明されており、その各フィールドの配置は以下のとおりである。

表7-5 mstatus register definition

Bit	Name	Access	Description	Reset value
[31:25]	Reserved	MRO	Reserved	0
24	MPPOP	MRW	Whether the current subactive interrupt needs to come out of the stack	0
23	MPOP	MRW	Whether the current active interrupt needs to come out of the stack	0
[22:13]	Reserved	MRO	Reserved	0
[12:11]	MPP	MRW	Privileged mode before entering break	0
[10:8]	Reserved	MRO	Reserved	0
7	MPIE	MRW	Interrupt enable state before entering interrupt	0
[6:4]	Reserved	MRO	Reserved	0
3	MIE	MRW	Machine mode interrupt enable	0
[2:0]	Reserved	MRO	Reserved	0

MPOPフィールドは、現在アクティブな割り込みがスタックから出る必要があるかどうかを制御・示すために使用される。値が1の場合はスタックから出る必要があり、0の場合は不要である。

MPPOPフィールドは、次にアクティブとなる割り込みがスタックから出る必要があるかどうかを制御・示すために使用される。値が1の場合はスタックから出る必要があり、0の場合は不要である。

通常、ネストが発生した場合、MPPOPはMPOPに更新され、MPOPは現在の割り込みのスタック出力フラグに更新される。ネストから復帰する際には、MPOPはMPPOPに復元される。

MPPフィールドは、例外または割り込みに入る前の特権モードを保存し、例外または割り込みから復帰する際に特権モードを復元するために使用される。QingKe V2マイクロプロセッサはMachineモードのみをサポートしており、MPPの更新機構は上記のような処理であるが、V2では値が0b11のみとなる可能性がある。

MIEはグローバル割り込み許可ビットであり、例外または割り込みに入る際、MPIEの値はMIEの値に更新される。QingKe V2シリーズマイクロプロセッサでは、Machineモードでの割り込みネストを継続して実行できるようにするため、最終レベルの割り込みに入る前にMIEは0に更新されない点に注意すること。

例外または割り込みから復帰した後、マイクロプロセッサはMPPに保存されたMachineモードに戻り、MPIEはMIEの値に復元される。

Hardware Instruction Set Register (misa)

このレジスタは、マイクロプロセッサのアーキテクチャおよびサポートされている命令セット拡張を示すために使用される。それぞれの項目についての定義は以下に記述される。

表7-6 misa register definition

Bit	Name	Access	Description	Reset value
[31:30]	MXL	MRO	Machine word length 1:32 2:64 3:128	1
[29:26]	Reserved	MRO	Reserved	0
[25:0]	Extensions	MRW	Instruction set extensions	x

MXLは、マイクロプロセッサのワード長を示すために使用される。QingKe V2は32ビットマイクロプロセッサであるため、このフィールドの値は1に固定されている。

Extensionsは、マイクロプロセッサがサポートする命令セット拡張の詳細を示すために使用され、それぞれが拡張の分類を表す。詳細な定義は以下の表に示される。

表7-7 Instruction Set Extension Details

Bit	Name	Description
0	A	Atomic extension
1	B	Tentatively reserved for Bit-Manipulation extension
2	C	Compressed extension
3	D	Double-precision floating-point extension
4	E	RV32E base ISA
5	F	Single-precision floating-point extension
6	G	Additional standard extensions present
7	H	Hypervisor extension
8	I	RV32I/64I/128I base ISA
9	J	Tentatively reserved for Dynamically Translated Languages extension
10	K	Reserved
11	L	Tentatively reserved for Decimal Floating-Point extension
12	M	Integer Multiply/Divide extension
13	N	User-level interrupts supported
14	O	Reserved
15	P	Tentatively reserved for Packed-SIMD extension
16	Q	Quad-precision floating-point extension
17	R	Reserved
18	S	Supervisor mode implemented
19	T	Tentatively reserved for Transactional Memory extension
20	U	User mode implemented
21	V	Tentatively reserved for Vector extension
22	W	Reserved
23	X	Non-standard extensions present
24	Y	Reserved
25	Z	Reserved

例: QingKe V2A マイクロプロセッサでは、このレジスタの値が `0x40800014` の場合、サポートされている命令セットアーキテクチャは `RV32EC` であり、非標準拡張 `X` も含まれている。また、Userモードの実装は存在しないことを意味する。

Machine Mode Exception Base Address Register (mtvec)

　このレジスタは、例外または割り込みハンドラのベースアドレスを格納するために使用される。また、下位2ビットは、セクション3.2で説明されているベクタテーブルのモードおよび識別方式を設定するために使用される。

Machine Mode Staging Register (mscratch)

表7-8 mscratch register definitions

Bit	Name	Access	Description	Reset value
[31:0]	mscratch	MRW	Data storage	0

　このレジスタは、Machineモードにおける一時的なデータ保存用の32ビット読み書き可能レジスタである。
　例えば、例外または割り込みハンドラに入る際、ユーザスタックポインタ(SP)の値をこのレジスタに保存し、割り込み用スタックポインタをSPレジスタに設定する。例外または割り込みから復帰する際には、mscratchからユーザスタックポインタSPの値を復元する。
　すなわち、割り込みスタックとユーザスタックを分離することが可能である。

Machine Mode Exception Program Pointer Register (mepc)

表7-9 mepc register definitions

Bit	Name	Access	Description	Reset value
[31:0]	mepc	MRW	Exception procedure pointer	0

　このレジスタは、例外または割り込みに入る際にプログラムポインタを保存するために使用される。例外または割り込みが発生した際に、例外に入る前の命令PCポインタを保存し、例外または割り込み処理後の復帰アドレスとしてmepcが使用される。

　ただし、以下の点に注意すること:

- 例外が発生した場合、mepcは現在例外を発生させている命令のPC値に更新される。
- 割り込みが発生した場合、mepcは次の命令のPC値に更新される。

　例外処理後に例外から復帰する必要がある場合、MEPCの値を適切に修正することに留意すること。詳細はChapter 2 例外 を参照。

Machine Mode Exception Cause Register (mcause)

表7-10 mcause register definition

Bit	Name	Access	Description	Reset value
31	Interrupt	MRW	Interrupt indication field 0: Exception	0

			1: Interruption	
[30:0]	Exception Code	MRW	Exception codes, see Table 2-1 for details	0

このレジスタは、例外の原因または割り込みの割り込み番号を格納するために主に使用される。

最上位ビットはInterruptフィールドであり、現在の発生が例外か割り込みかを示すために使用される。

下位ビットは例外コードであり、具体的な原因を示すために使用される。

詳細についてはChapter 2 例外 を参照すること。

Debug Control and Status Register (dcsr)

このレジスタは、デバッグモードの動作状態を制御および記録するために使用される。詳細な説明についてはセクション6.1を参照すること。

Debug Mode Program Pointer Register (dpc)

このレジスタは、マイクロプロセッサがデバッグモードに入った後に実行される次の命令のアドレスを格納するために使用される。デバッグモードに入る理由によって、このレジスタの値は異なる規則で更新される。詳細についてはセクション6.1を参照すること。

Debug Mode Staging Register (dscratch0-1)

このレジスタ群は、デバッグモードにおける一時的なデータ保存のために使用される。

表7-11 dscratch0-1 register definitions

Bit	Name	Access	Description	Reset value
[31:0]	dscratch	DRW	Debug mode data staging value	0

7.3 User-defined CSR Registers

Interrupt System Control Register (intsyscr)

このレジスタは、割り込みネストの深さ、HPE、EABIの有効化などの関連機能を設定するために主に使用される。詳細な説明についてはセクション3.2を参照すること。